L2: Entry 1 of 3 File: USPT Aug 29, 2000

DOCUMENT-IDENTIFIER: US 6112241 A
TITLE: Integrated network interconnecting device and probe

Abstract Text (1):
A Local Area Network (LAN) Switch includes conventional switching functions and integrated Remote Monitoring (RMON) Universal Feature Card (UFC). The UFC allows simultaneously statistically monitoring the traffic on all ports, 100% monitoring of the traffic on one port, and monitoring the internal LAN Switch switching fabric to obtain RMON statistics about the operation of networks attached to the LAN Switch.

Brief Summary Text (3):
The invention relates to computer networks in general and, in particular, to devices and methods that monitor such networks.

Brief Summary Text (6):
Remote monitoring (hereafter called RMON) of LANs for problem isolation and determination has always been necessary to ensure proper operation of the LAN. Usually, the monitoring is done at a customer premises.

Brief Summary Text (7):
Initially, RMON devices were used to measure the physical parameters of the networks. Eventually, ISO layer 2 devices were used to capture a series of data bits, on the LAN, that were interpreted by the user. Each user had different requirements and each device had to be configured to measure the desired parameters and data pattern. To facilitate consistency in monitoring, the Remote Monitoring (RMON) standard (Request for Comments--RFC 1757) has been developed to provide a standard set of data parameters to be gathered and displayed on a network management workstation to determine network operation. For this information to be obtained, a device called a probe had to be attached physically to the network to be monitored. For single segment LANs, the probe can easily be attached to collect the RMON and display RMON information.

Brief Summary Text (10):
Still other examples of the prior art techniques and systems are set forth in U.S. Pat. Nos. 5,231,593; 5,251,152; 5,101,402 and articles entitled "Axon Tackles Switch Traffic Monitoring" and "Switches Integrate Monitoring" by Claudia Graziano, LAN Times (Online).

Brief Summary Text (11):
Even though the above prior art works well for their intended purposes, they are not effective when used in networks interconnected by interconnecting devices such as switches. In switch connected networks, multiple probes are required to monitor the segments. Adding probes to each segment is very costly and difficult to manage. Another problem is that in some of the prior art, such as U.S. Pat. No. 5,101,402, the statistics are gathered at the session level. The name "session level" suggests that the statistic gathering is done at layers above the layer 1 (physical) and layer 2 (data link) of the International Standard Organization (ISO) seven layer model. It is believed that gathering layer 1 and layer 2 (physical and data link) statistics are necessary for effective network management.

Brief Summary Text (12):
Still another problem which the prior art does not even recognize, much less address, is the gathering of RMON statistics and display RMON information about the internals of the switch. Such internal monitoring can be of immense importance in managing and distributing load in the network. The present invention provides internal monitoring of the switch.

Brief Summary Text (18):
It is still another object of the present invention to monitor the switch internally and provide internal RMON statistics.

Brief Summary Text (20):
These and other objects of the invention are achieved by a RMON system termed (RMON) Universal Feature Card (UFC), having a processor sub-system coacting with a statistical gathering subsystem that is provided with dedicated port monitoring functions that gather statistical data from any one of the device ports on a fixed time basis or gather statistical data on a time-sliced basis (roving) from all the ports. The time-sliced information gathering is termed "Roving". In addition, the internal bus or other switch fabric of the switch is monitored and statistical information on internal bus operation is provided. The remote monitoring and information gathering are done simultaneously.

Brief Summary Text (21):
In particular, the RMON system allows the user or customer to gather statistics on all ports simultaneously in a statistical manner. It also allows the customer to monitor, in real-time mode, all the traffic on one segment. With the time-slice statistical gathering of RMON statistics (called roving), the user can determine where the problem occurs. With the fixed monitoring of real-time traffic (called dedicated), the user can isolate the problem because the dedicated port gathers all the traffic on the segment in question. The RMON system also allows the user to look at the cross-port traffic internal to the LAN Switch and obtain performance data so that one knows how to segment the domains and have the most efficient segmentation of one's network.. The internal monitoring also provides statistics on high speed LANs called "Uplinks" that may be connected to the switching fabric.

Brief Summary Text (23):
In any given product, the user can monitor the RMON functions on all ports. When one port has a problem, the RMON UFC's program can then fully monitor the port in question while continuing to monitor the RMON statistics on the others. If the customer wants to monitor one port in full-duplex mode, the selection circuitry can be set to monitor both the transmit and receive traffic on one port only in dedicated mode. In this scenario, the roving function is not available. Since servers will usually be connected to the full duplex ports, this allows the customer to get a picture of the server traffic patterns and utilization levels.

Drawing Description Text (5):
FIG. 4 shows a Programmable Logic Device (PLD) that selects ports to be monitored on a fixed bases or time shared basis (roving) according to the teachings of the present invention.

Detailed Description Text (5):
Still referring to FIG. 1, the LAN Switch 10 includes remote monitor (RMON), Universal Feature Card (UFC) "1--1" and ATM Universal Card "2-1". The ATM card is connected to the Switching Fabric and Port 2-1. The function of the ATM UFC is to provide a high speed uplink so that stations and/or LAN segments connected to LAN switches can communicate via the high speed network. It should be noted that other types of high speed networks, such as 100 Mbps Token Ring and/or 100 Mbps Ethernet feature card could be inserted instead of the ATM. As discussed above, the presence of a

Detailed Description Text (7):
Still referring to FIG. 1, the RMON UFC "1--1" provides remote monitoring in LAN Switch 10. The integration of the remote monitor, also called a probe, to the best of applicants' knowledge, has never been done, before, in the switch. By integrating the probe (RMON UFC) in the switch, several benefits and functions described below are provided which were not available in prior art switches. Optionally, the RMON UFC can be connected to a RMON Port 18 with a connector. Because the port is optional, it is shown in broken lines. This RMON port can be used for communicating directly with the RMON UFC for monitoring information in the RMON UFC. It should be noted that without the port, communication with the RMON can be effectuated through any of the other connector ports on the switch.

Detailed Description Text (10):
FIG. 2 shows a block diagram of the RMON UFC, known also as Probe, according to the teachings of the present invention. It should be noted that RMON UFC and Probe are used interchangeably to represent the same device in the present invention. The RMON UFC includes RMON UFC Connector 24 (FIG. 3), Programmable Logic Device (PLD) 26, Circuit Arrangement 28 and RMON Processor

Subsystem 30. The RMON Connector 24 (FIG. 3) collects signals from the receive/transmit (XMIT) clock lines and receive/transmit (XMIT) data lines. The RMON Connector can be as elaborate as one chooses or simple taps as is shown in FIG. 3. The Programmable Logic Device 26 monitors the receive/transmit data and receive/transmit clock lines from all switch ports and selects one for future processing. The Circuit Arrangement 28 includes the hardware for gathering of data from the port interface and the switch fabric interface. The RMON Processor Subsystem 30 includes the software which process the gathered information to generate RMON statistics and other information which is used for managing the network. In the preferred embodiment of this invention, a i960 microprocessor fabricated by the Intel.RTM. Corporation is used. For purposes of discussion, it is also assumed that the main LAN Switch Processor is also an i960 processor. The bus of both processors are coupled via the Mailbox 32. The function of the Mailbox 32 is to enable the interchange of information between the LAN Switch Processor and the RMON Processor. It should be noted that any other processor, other than the i960, can be used without deviating from the scope or spirit of the present invention.

Detailed Description Text (12):
It should be noted that signals, hereinafter called traffic, can be monitored on both Half-Duplex Ports and Full-Duplex. The monitoring is as follows:

Detailed Description Text (14):
In RX-mode: All traffic and a port, say Port X, is monitored. This means traffic from Port X addressed to hosts on other segments and hosts on the segment. Also, all traffic to Port X from other hosts or segments is monitored.

Detailed Description Text (15):
In TX-mode: This mode is not available since all traffic is monitored.

Detailed Description Text (17):
In RX-mode: All traffic from Port X addressed to other ports and the other segments are monitored.

Detailed Description Text (18):
In TX-mode: All traffic to Port X from host to other segments are monitored.

Detailed Description Text (19):
In addition to either full or half duplex ports, the user may elect to monitor a port in the DEDICATED mode or ROVING mode. With reference again to FIG. 2 and FIG. 4, in the Dedicated mode, the Dedicated Multiplexer 26' selects the designated port. Likewise, if the roving mode is selected, the Roving Multiplexer elects the port to be monitored. As is used in this document, dedicated mode means that the user obtains all the traffic from a selected port and the RMON statistics are 100% accurate. Roving mode means that the gathering of RMON statistics are done in a time slice manner. Statistics are gathered for a certain time on one port, the monitored port is changed and statistics are gathered on the new port for a certain period. The program and hardware statistically rove all the ports on the LAN switch to gather enough information to provide RMON statistics. The gathered statistics are then adjusted to compensate for the time slice mode of gathering information. The adjustment, for example, increases a port's statistics by eight if eight ports are currently being roved in an equal time slice. If traffic load on a port is high (relative to other monitored ports), the roving logic may spend more time gathering statistics on this port and, therefore, the adjustment would not have an equal uplift as the other lightly loaded ports.

Detailed Description Text (20):
Still referring to FIG. 2, the Circuit Arrangement 28 includes Mailbox 32, that interconnects the i960 Processor Bus of the LAN switch with the i960 Processor Bus of the RMON UFC. Commands are passed between the LAN Switch i960 Processor and the RMON UFC i960 Processor through the Mailbox 32. Appendix B sets forth a list of commands used between the processors. Other commands can be used without deviating from the present invention. The Dedicated Internal Monitor 34 (including Network Management and Monitoring) is connected to Switch Fabric 12, over appropriate busses and through appropriate busses and Tri-state Circuitry 35 to the i960 Processor Bus. The Dedicated Internal Monitor System 34 (details given below) provides an internal interface for monitoring traffic across the Switch Fabric 12. This function provides RMON 1 and RMON 2 statistics for the switch fabric and port within the LAN switch. Among other things, the dedicated internal monitor provides RMON function for high speed LAN switch (uplinks) for which no RMON information could previously be gathered. With respect to FIG. 1,

the information gathered would be relative to the ATM high speed uplink.

Detailed Description Text (22):
Referring now to FIGS. 2 and 5 (wherein FIG. 5 shows a more detailed block diagram of Circuit Arrangement 28 and RMON Processor Subsystem 30) the Dedicated Internal Monitor 34 includes Network Management Universal Feature Interface Chip (UFIC) 44 and Monitoring UFIC 46. The UFICs 44 and 46 are connected by a UFIC Bus 48 to Data Storage 50 and Tri-state Circuit 35. The Tri-state Circuit 35 isolates the UFIC Bus 48 from the i960 CF Processor Bus 84. Access to use the Bus 84 or Data Storage 50 is provided by the Arbitration/Memory Control Logic Circuitry 52. In the preferred embodiment of this invention, the Data Storage 50 is an SRAM storage with approximately 512 Kbytes. Also, the Mailbox 32 connected to UFIC 44 and the i960 CF Processor is a two port random access memory device (RAM).

Detailed Description Text (23):
Still referring to FIG. 5, the Network Management UFIC 44 is used for monitoring normal communications to/from the RMON UFC and the network management workstation containing the RMON display application and connected to one of the switch ports or somewhere else in the network. With respect to the Network Management UFIC 44, initially the location of the Management Station is not known. As a message enters the LAN switch for the first time, it is broadcast to all outputs on the switch fabric. In the case of the RMON UFC, this message is received by the Network Management UFIC 44 and placed in Data Storage 50. An interrupt sent to i960 CF Processor then fetches this message, processes it and formats a response. The response is placed in Memory 50 and the UFIC automatically appends switch fabric routing information so the message is sent to the appropriate port containing the network management station. Once a connection between the RMON UFC and the network management station is established, communication is direct unicast traffic without using the broadcast messaging capability of the switch.

Detailed Description Text (24):
Still referring to FIG. 5, the Monitoring UFIC 46 is useful to monitor switch fabric traffic into or out of one of the "FAT-PIPE" (high speed link) UFCs, such as the ATM UFC 2-1 FIG. 1). To perform this function, the Monitoring UFIC 46 (to be described subsequently in greater detail) contains a monitor register set by a LAN switch program (to be discussed later) executing on the i960 LAN Switch Processor. First, the LAN Switch Processor sends a table of all the switch interfaces to the RMON UFC at the end of the initialization process. The user then decides which interface to monitor and sends this request to the RMON UFC. Upon command of the RMON UFC, the LAN Switch i960 processor sends a message to the Monitor UFIC 46 to monitor traffic on the switch fabric from a specific high speed port. The Monitor UFIC 46 then "listens" on the switch fabric for messages with addresses matching that of the specific high speed port. When a match is found, the message is copied from the switch fabric into the Data Storage 50 and sent to the RMON UFC processor for processing.

Detailed Description Text (37):
Still referring to FIG. 7, and in particular Block 112, the program also descends into Block 116 where data associated with a dedicated port and data associated with a roving port are collected. The program then descends into Block 118 where a 100 millisecond timer is started. The time represents the period for which data is monitored on a port, using the roving function of the invention. The program on the RMON Processor then descends into Block 120 where it tests to see if the time set in Block 118 has expired. If it has not, the program loops until it is expired and descends into Block 122 where the RMON Processor stops processing roving data while processing of the dedicated data is continued. The program then descends into Block 124 where processing on the Dedicated Port 1 still continues while the program selects the randomly roving Port N. Any messages from Block 124 are stored in the mailbox and are sent to the LAN switch program Block 126, where the program is set to Dedicated Port 1 and Roving Port N. In FIGS. 7 and 8, B, preceding all numerals, is the abbreviation for Block.

Detailed Description Text (40):
The RMON UFC according to the teachings of the present invention provides a Remote Network Monitoring, MIB consistent with RFC 1757 Standard, incorporated herein by reference. A set of RFC 1757 Statistical Groups provided by the invention is set forth in Table 2 of Appendix A.

Detailed Description Text (41):
Because there are contentions for the RMON i960 Processor and Memory, the base RMON i960 Card contains a arbitration mechanism to ensure that all tasks can be executed in the most efficient.

manner possible. To this end, there is a priority scheme established for the interfaces. The UFIC Monitoring Interface has the highest priority and always gets service. The Dedicated and Roving Port interfaces operate at a much slower speed and have second priority. These two are serviced equally in this priority level. If one port is being serviced and both ports have an interrupt raised, the arbitration mechanism services the one not currently being serviced to ensure fairness. The lowest priority is the network management interface UFIC. This interface is not time constrained like the other three. The following Table 1 illustrates the priority of the different interfaces, preferably implemented on a daughter card.

Detailed Description Text (43):
In addition to the single port roving and dedicated RMON function, the RMON UFC is capable of providing RMON function for traffic on the internal LAN Switch. To provide this function, the Monitoring UFIC 46 is configured in the monitor mode. The configuration is done by the LAN Switch i960 Processor to monitor data traffic from a specific port. In this mode, the UFIC receives data from the internal LAN switch bus, it stores this data in Data Storage 50. The LAN Switch i960 processor is responsible for initializing the control blocks for storing the data. Once an entire frame has been stored, the UFIC chip interrupts the RMON i960 processor and it services the frame. When the servicing is completed, the RMON i960 Processor 82 releases these buffers as free buffers in the buffer descriptor list. Each time a frame has been received in the memory, the RMON i960 processor is interrupted. The interrupt microcode must contain a tight timing loop since it could receive multiple interrupts for the small frames at 160 Mbps. For a 64-byte message, the RMON i960 has approximately 7 microseconds to process the interrupt before the next one arrives.

Detailed Description Text (47):
The RMON UFC provides statistically accurate RMON groups on all LAN Switch interfaces. When a problem is found, the user can then turn his attention to the questionable interface by dedicating the gathering of RMON statistics to one interface while continuing to simultaneously monitor all LAN Switch interfaces. Dedicating the collecting of RMON groups allows the customer to investigate and solve the problem.

Detailed Description Text (48):
The user also obtains a picture of the activity inside the switch with the RMON UFC's internal monitoring of traffic on the switch fabric. With this function, the user can see segment-to-segment problems and re-route or rearrange his traffic patterns inside the LAN Switch.

Detailed Description Paragraph Table (1):

TABLE 1	TASK	PRIORITY	OF SERVICE
RMON i960 Processor	1	UFIC Monitor traffic	2
UFIC Network Management Traffic	Dedicated port message complete	3	Roving port message complete

Detailed Description Paragraph Table (15):

APPENDIX B	Command Direction	Type	Comments
CONFIGURE SECONDARY	to LAN Switch	Unsolicited	UFIC configuration table to be UFIC set by the LAN Switch i960. SET TCP/IP ADDR to RMON UFC	Command/Response	Set TCP/IP Address (may not need this)
START DEDICATED PORT to LAN Switch	Command/Response	Mux set on LAN Switch to monitor dedicated port. Begin to gather stats.
START DEDICATED PORT to RMON UFC	Command/Response	Acknowledge reception of ACK command
START ROVING PORT to LAN Switch	Command/Response	Mux set on LAN Switch to monitor roving port. Begin to gather stats.
START ROVING PORT to RMON UFC	Command/Response	Acknowledge reception of ACK command
START MONITOR BUS to LAN Switch	Command/Response	UFIC set to monitor LAN Switch internal bus port. Begin to gather stats.
START MONITOR PORT to RMON UFC	Command/Response	Acknowledge reception of ACK command
GET PORT INFO to LAN Switch	Command/Response	Obtain the LAN Switch port configuration tables
GET PORT INFO ACK to RMON UFC	Command/Response	Indicate the LAN Switch port configuration tables have been loaded in the mailbox
PORT STATUS CHANGED to RMON UFC	Unsolicited	Indicate the LAN Switch port configuration tables have changes are loaded in the mailbox

Other Reference Publication (1):
"Axon Tackles Switch Traffic Monitoring" by Claudia Graziano, Lantimes Online Apr. 24, 1995.

CLAIMS:

7. The network interconnecting device of claim 6 wherein the probe further includes an interface that monitors information on the switching fabric;

a second circuit arrangement for processing monitored information to extract data signals representative of selected ones of the monitored information; and

storage, accessible to the controller, for storing the data signal.

12. A probe for monitoring networks coupled by an interconnecting device including:

a circuit arrangement, mounted inside the interconnecting device, that collects signals on a fixed time basis manner or time-slice manner at ports of said interconnecting device;

a logic circuit arrangement that processes the signals and generating statistical information therefrom; and

a processor based system that processes said statistical information and arranging them into RMON data groups as specified by RFC 1757 standard.

15. A probe for integrating in a network interconnecting device including:

a first interface with circuits that monitor signals from at least one port within a group of ports to collect data on a fixed time basis from said at least one port;

a second interface with circuits that roves the ports in the group and monitors signals to collect data on a time-slice basis; and

a controller that processes the data and arranges the processed data in a Management Information Base (MIB).

18. The probe set forth in claim 15 further including a third interface that monitors internal operations of the network interconnect device to gather internal statistics.

Previous Doc Next Doc Go to Doc#

L2: Entry 2 of 3 File: USPT Aug 22, 2000

DOCUMENT-IDENTIFIER: US 6108782 A
TITLE: Distributed remote monitoring (dRMON) for networks

Abstract Text (1):
Distributed remote monitoring (dRMON) of network traffic and performance uses distributed nodes to collect traffic statistics at distributed points in the network. These statistics are forwarded to collectors which compile the statistics to create combined views of network performance. A collector may mimic a prior art, non-distributed, network probe and may interact with network management software as though it were a stand alone network probe thereby simplifying a user's interaction with the distributed system. The invention is designed to work in accordance with a variety of standard network management protocols including SNMP, RMON, and RMON2 but is not limited to those environments. The invention has applications in a variety of communication system environments including local area networks, cable television distribution systems, ATM systems, and advanced telephony systems. A specific embodiment of the invention solves is particularly optimized to work in LAN environments with end systems running under Windows-compatible network operating systems.

Brief Summary Text (4):
This invention relates to transmission of information between multiple digital devices on a network. More particularly, this invention relates to a method and apparatus for monitoring and analysis of network traffic using a distributed remote traffic monitoring (DRMON) technology.

Brief Summary Text (8):
This specification also presumes some familiarity with the specific network and operating system components discussed briefly in the following paragraphs, such as the simple network management protocol (SNMP) for management of LAN and WAN networks, and the RMON MIBs defined for remote network monitoring and management.

Brief Summary Text (23):
Management and Monitoring of Individual ESs in a Network Environment

Brief Summary Text (24):
A network such as that shown in FIG. 1 is generally managed and monitored within an enterprise by a central Information Services department (ISD), which is responsible for handling all the interconnections and devices shown. The same ISD is generally responsible for managing the applications and system components on each of the individual ESs in the network.

Brief Summary Text (25):
Many prior art systems have been proposed to allow an IS staff person to manage and partially monitor network infrastructure remotely over a network. Such systems include IBM's NetView, HP's OpenView or Novell's Network Management System (NMS). However, these systems generally rely on a full network protocol stack to be correctly running effectively on the remote ES in order to accomplish any remote file management operations.

Brief Summary Text (27):
A common protocol used for managing network infrastructure over the network is the Simple Network Management Protocol (SNMP). SNMP is a layer 7 network and system management protocol that handles network and system management functions and can be implemented as a driver (or SNMP agent) interfacing through UDP or some other layer 4 protocol. Prior art SNMP installations largely were not placed in ESs because SNMP did not handle ES management or monitoring functions and because SNMP agents are processor and memory intensive.

Brief Summary Text (28):

SNMP is designed to provide a simple but powerful cross platform protocol for communicating complex data structures important to network infrastructure management. However, its power and platform-independent design makes it computationally intensive to implement, and for that reason it has limited applications in end system management or monitoring. It is primarily used in network infrastructure management, such as management of network routers and bridges.

Brief Summary Text (31):
SNMP is described in detail in a number of standard reference works. The wide adoption of SNMP throughout the networking industry has made compatibility with SNMP an important aspect of new management and monitoring tools.

Brief Summary Text (33):
Prior art Remote Monitoring (RMON) technology is a set of software and hardware specifications designed to facilitate the monitoring and reporting of data traffic statistics in a local area network (LAN) or wide area network (WAN). RMON was originally defined by the IETF (Internet Engineering Task Force) in 1991. RMON defined an independent network probe, which was generally implemented as a separate CPU-based system residing on the monitored network. Software running on the probe and associated machines provided the various functions described by the defining IETF RFC documents, RFC-1271, RFC-1513 and RFC-1757.

Brief Summary Text (36):
(1) RMON provides autonomous Network Management/Monitoring, unlike SNMP which required periodic polling of ESs. RMON stand-alone probes are constantly on duty and only require communication with a management application when a user wishes to access information kept at the probe.

Brief Summary Text (39):
(4) RMON permits the collection and maintenance of historical network performance metrics thereby facilitating trend analysis and proactive performance monitoring.

Brief Summary Text (41):
The new capabilities of RMON were quickly appreciated and RMON probes soon became the preferred choice for remote monitoring. It has become common place for ISs, particularly hubs and switch/bridges to embed RMON probe functions.

Brief Summary Text (43):
Shortly after adoption of RMON, users wanted more management information than the layer 2 statistics RMON provided. In particular, network managers wanted to track higher layer protocols and the sessions based upon those protocols to learn which applications were using which protocols at what expense in available network bandwidth. Therefore, a new version of RMON, RMON2 was developed to provide more advanced capabilities. RMON2 provides network header layer (layer 3) through application layer (layer 7) monitoring for a number of commonly used protocols and applications, including the Internet protocol suite (IP and UDP) and Internet applications (FTP, Telnet, TCP and SNMP).

Brief Summary Text (45):
A traditional stand-alone RMON probe, connected to a switch like any other host device, only sees network traffic flowing on the segments to which it is connected, greatly limiting its usefulness in modern, more complicated network topologies. One solution is to place the RMON probe within the switch itself and have it monitor all ports simultaneously. However, this requires considerable processing capability in order to handle the large bandwidth made possible by modern switching architectures.

Brief Summary Text (46):
In a conventional 10 Mb Ethernet or 4/16 Mb Token Ring environment, a stand-alone RMON probe on a single network segment could usually be implemented on a 486-class processor. However, where multiple network interfaces must be monitored or where network bandwidths are higher, (such as with 100Base-T LANs or switching hubs/ATM), it is considerably more costly to build a probe with sufficient processing power to capture all, or even most, of the network packets being exchanged. Independent laboratory tests show that RMON products claiming to keep up with higher bandwidth network traffic generally cannot, in fact, keep up with all data flow during peak network rates. The situation worsens considerably when attempting to do RMON2 analysis of network packets in high bandwidth environments. Processing power required can be easily five times greater than needed to simply capture packets, and data storage requirements can easily increase ten fold.

Brief Summary Text (47):
Use of filtering switches and hubs (discussed in the above referenced patent applications) in networks further limits the usefulness of probes because, unlike repeaters, not all the packets appear at every output port of the switch. This makes the use of external stand-alone probes infeasible unless the switch vendor has provided a monitor port (sometimes called a copy port) where all packets are repeated to the external RMON probe. However, this approach decreases data traffic performance in the switch, and does nothing to reduce the processing overhead required of the probe.

Brief Summary Text (49):
For purposes of clarity, the present discussion refers to network devices and concepts in terms of specific examples. However, the method and apparatus of the present invention may operate with a wide variety of types of network devices including networks and communication systems dramatically different from the specific examples illustrated in FIG. 1 and described below. It should be understood that while the invention is described in terms of a computer network, the invention has applications in a variety of communication systems, such as advanced cable television systems, advanced telephone networks, ATM, or any other communication system that would benefit from distributed performance monitoring and centralized collection and compilation. It is therefore not intended that invention be limited, except as indicated by the appended claims. It is intended that the word "network" as used in the specification and claims be read to cover any communication system unless the context requires otherwise and likewise "end system" and "node" be read to encompass any suitable end system (telephone, television) on any such communication system or to encompass distributed points in the network intermediate of an end systems. It is also intended that the word "packet" as used in the specification and claims be read to cover any unit of transmitted data, whether an ethernet packet, a cell, or any other data unit transmitted on a network unless the context requires otherwise.

Brief Summary Text (51):
The present invention is a method and apparatus for distributed remote network monitor (dRMON) in a LAN. According to an embodiment of the invention, dRMON agents, which are software or software plus hardware components, are placed within each (or a subset) of the ESs such as 50a-c, 51a-c, and 521-g, connected to the LAN or within server machines. These agents implement prior art RMON functional groups but only capture and analyze packets that their native ES sends or receives, or in some embodiments captures packets that the ES communicates with an ES that does not have an dRMON agents installed; as a result, the processing requirements of the dRMON agents are kept well within the range of the ES (or host) CPU's capabilities and generally do not result in a noticeable loss of performance.

Brief Summary Text (53):
According to one embodiment of the invention, a dRMON collector can mimic the SNMP responses of a prior art non-distributed RMON probe so that existing network management or monitoring software can interact with the collector as though the collector were a prior art probe. Therefore prior art network management software need not be aware of the existence of the dRMON agents.

Brief Summary Text (54):
According to a further embodiment, multicast domains are handled specially. In a default mode, ESs in the same multicast domain are treated by a collector as though they are on one LAN segment. This approach allows other vendor's RMON network management applications to interact with the collector as though it were a prior art probe; however, when used with enhanced dRMON Managers, a user is provided the ability to combine ports and hosts in order to create Virtual LAN (VLAN) definitions which would cause the monitoring function to behave as though all selected hosts were on the same LAN segment being served by the same RMON probe. A dRMON collector in this embodiment could create and maintain several such views with each appearing as one interface to conventional RMON Management applications.

Brief Summary Text (57):
There are several key advantages to various embodiments of the invention when compared to other solutions. among these advantages are scalability, affordability, true end-to-end response time monitoring, redundancy, visibility into client node, distributed architecture, and web support.

Brief Summary Text (58):

Because each agent is analyzing only its own directed traffic, or possibly its own traffic and the traffic of a limited number of other ESs, dRMON can handle extremely high bandwidth environments with relative ease. Compared to stand-alone probes, dRMON is more affordable as a remote monitoring tool, particularly in switched environments. Very inexpensive PC technology can be used to host the Collector software resulting in low equipment costs.

Brief Summary Text (59):
RMON2, for all its power, still does not afford the network manager one of the most asked for features, that being continual response time monitoring. RMON2 applications can only do this if packet capture is used to forward the protocol streams to the management station, at a price in network utilization and performance. dRMON Agents routinely perform this analysis and forward the results (not the entire packets) to the Collector.

Brief Summary Text (61):
Since data collection is done by the managed nodes and RMON Collectors can substitute for each other, there is no single point-of-failure and dRMON therefore inherently provides monitoring redundancy. In the case of monolithic probes or management add-in cards, unless multiple probes are deployed on each LAN segment, a probe's failure can be disastrous when attempting remote monitoring.

Detailed Description Text (2):
FIG. 1 is a block diagram illustrating the deployment of the invention in an example network according to a specific embodiment of the invention. The invention includes two types of primary components, the agents that reside in ESs and the collector or collectors that collect and compile the network statistics and interacts with network management applications (such as an application running on console 54) to provide a management/monitoring picture to the network.

Detailed Description Text (5):
FIG. 4 shows one particular embodiment of an agent and other components upon which it depends. An NDIS DeskTop Agent type module (DTA) is used to bind to the network adapter driver, thus establishing a source of directed packets to analyze as well as a means to communicate with the dRMON collector via the network. Multiple NIC bindings may be supported by the agent and may allow the agent to monitoring traffic on different segments having different layer 1 protocols.

Detailed Description Text (14):
While the invention may be most easily described as a network having a single collector, because the actual data gathering and monitoring is being performed at the managed ESs, it is possible to have another collector on the LAN/WAN assume the data collection duties of a defective or off-line collector. It is also possible to have multiple collectors on a LAN, in which case in this embodiment an identifier is used so that an agent communicates with only one collector. In one embodiment, this identifier also acts as a security password as described below.

Detailed Description Text (15):
FIG. 6 is a block diagram of an embodiment of a dRMON Collector according to the invention. Like the Agent, the Collector loads automatically when the system starts and depends upon the same DTA services to exchange dRMON protocol traffic with its Agents. The DTA is also used as a packet interface to allow the Collector to monitor its own directed traffic as well as the broadcast and multicast traffic flowing within its sphere of management. To prevent duplication of statistics, only the Collector maintains RMON information on broadcast and multicast traffic.

Detailed Description Text (18):
The Integrator 148 merges RMON statistics, tables and capture streams coming from the remote dRMON agents with the equivalent output from the Collector's analysis of its own directed traffic combined with the broadcast and multicast traffic present at its interface. The final result is an integrated view of all of the monitored traffic just like one would get from a conventional RMON probe.

Detailed Description Text (25):
For the purposes of this description of the invention, we will refer to the protocol by which dRMON collectors and agents communicate over the network as the dRMON protocol. Unless the

context otherwise requires, the dRMON protocol should be understood to represent any possible protocol between collectors and agents for the exchange of management/monitoring information, generally in the form of MIBs, including prior art SNMP-type protocols or including a preferred specialized protocol as just described.

Detailed Description Text (27):
From the perspective of the user, the primary functions of the agents and the collector are to collectively implement the monitoring, management, and packet capture capabilities defined from RMON2, SNMP, and related networking standards with enhancements resulting from the distributed nature of the invention as herein described. As these primary functions are described in publicly available standards and documents and are well-know to practitioners in the art, details of the network statistics gathering, packet capture, or standards-based configuration of those

Detailed Description Text (28):
function are not described here. What follows is a description of the functions according to the invention that allows the invention to perform network monitoring, in a distributed fashion.

Detailed Description Text (74):
A special Transmit Callback from the dRMON Agent is supported in drivers designed for use with the invention. This transmit callback allows outbound traffic from the host to be monitored by dRMON without the performance penalty resulting from putting the adapter in promiscuous mode, as is currently required in many prior art drivers in order to see transmit traffic. In some current network operating systems there is no way for a higher layer protocol (such as the dRMON agent) to signal to the driver that it wants to see copies of data that is being transmitted on the network.

Detailed Description Text (75):
According to the invention, the dRMON agent performs a set operation against the NIC driver using the transmit callback OID, indicating a 32-bit pointer to the dRMON agent's call-back routine. If that operation succeeds, then the dRMON agent knows that the NIC driver includes code to support the transmit callback. The agent then can instruct the NIC driver, using set operations, to set NIC driver filters to monitor directed transmit traffic. If the callback set operation fails, then the agent sets the adaptor filters to promiscuous mode, in which case the adaptor reads all packets that appear on the wire, including packets it transmits, and those packets are available to higher layer protocols.

Detailed Description Text (98):
Domain Collectors (DCs) are used in larger networks to collect and archive management data from Workgroup Collectors within their sphere of management. DCs typically represent larger regions within the enterprise network such as a remote office or a whole building on a large campus. Each one can support multiple management stations 84, thus permitting any manager to monitor that domain from anywhere in the enterprise. Because of their greater scope of responsibility and the need to provide considerable long term and nonvolatile data storage, DCs are generally much more powerful devices than Workgroup Collectors and as such, are generally implemented as stand alone stackable devices generally located with the switches and hubs they oversee.

Detailed Description Text (102):
(1) Probe Based. RMON probes often have more resources available than do management cards embedded in switches and hubs and are often strategically located throughout the network in a way that makes them prime candidates for collection points for dRMON. Combined with a desire to monitor devices which do not have a dRMON agent installed, locating a Collector in the probe has further advantages. For example, a dual-interface RMON probe could be connected to two switch ports which are shared with a number of older PCs, Mackintoshes and UNIX workstations which do not have dRMON Agents. All other dRMON-equipped nodes would be distributed across the other switch ports. Ideally, the probe would be configurable to provide a choice of views such that the user could select to have the probe combine the Collector's data with its own to create one interface view or to present them as separate interfaces.

Detailed Description Text (104):
(3) Stackable/Stand alone. The Stackable Collector is a dedicated dRMON Collector whose packaging may be identical to that of the stackable hubs which it would manage. It may be based upon proprietary hardware or possibly a PC without monitor or keyboard. This Collector has a

more powerful CPU than most embedded management cards and is capable of holding considerable RAM and, optionally, hard disk storage; as a result, it can hold much more RMON data such as large amounts of historical data or captured packets. It may also provide additional services such as WEB-based RMON management and even WEB-based device management of the rest of the stack. The inclusion of many of these enhanced capabilities into this Collector's specifications are facilitated by basing it upon the PC architecture and using an OS such as Windows NT to support various add-ons. The development tools for the PC platform are also far ahead of those for embedded processors, thus shortening substantially the time-to-market and maximizing the availability of experienced programmers.

Detailed Description Text (106):
While dRMON Agents distribute RMON's functionality on the front-end (i.e. at the ES level), it is Domain Collectors 80 which distribute it on the back-end (i.e. at the management terminal level). DCs are generally implemented on powerful hardware, possibly based upon Pentium/Pentium Pro systems running Windows NT. DCs are concentrators for large amounts of network management data. In one embodiment, DCs allow capturing more network monitoring data without overly burdening distributed collectors by periodically off-loading statistics from the ISs, freeing up those IS resources to continue to capture new data. This data is gathered from a variety of possible sources, such as: dRMON Workgroup Collectors, Embedded RMON (full or partial) in switches/hubs, RMON probes and/or Embedded SNMP Management Agents in switches/hubs. A DC merges and organizes this various information to create a seemingly homogenous view of its management domain. The management domain may include different LANs that communicate across routers and domain collectors generally are able to communicate via a routed network protocol, such as IP. The merged view is then made accessible in any variety of possibly ways, including to compliant SNMP-based management applications, published using WEB protocols, via dial-up, etc. Because of the large and extensible storage capabilities that may be included with DCs, considerable historical data and many large captured packet streams could be maintained and archived and offered to any management station anywhere in the enterprise.

Detailed Description Text (113):
The invention may be embodied in a set of executable computer program code which may be stored into a fixed computer medium such as a disk, diskette, volatile memory or non-volatile memory, or any other medium for storing computer code. In such a case when such instructions are loaded and executed in an appropriately configured network intermediate system, the intermediate system will perform as described herein. A representation of such a system 700 in shown in FIG. 11, containing CPU 707, optional input devices 709 and 711, disk drives 715 and optional monitor 705. Fixed media 717 may be used to program such a system and could represent a disk-type optical or magnetic media or a memory. A system such as 700 may be used in conjunction with the invention as embodied on a fixed media to generate executable files that can be distributed throughout a network to various network components as described herein.

Detailed Description Text (114):
The invention has now been explained with reference to specific embodiments. Other embodiments will be apparent to those of skill in the art. In particular, method steps have been grouped and labelled as being part of various sub-methods in order to increase clarity of the disclosure, however, these steps could be differently grouped without changing the essential operation of the invention. Furthermore, it should be understood that while the invention has been described in terms of a computer network, the invention has applications in a variety of communication systems, such as advanced cable television or telephone networks, or any other communication system including system performance monitoring at distributed points in the system and reported back to a centralized collector. It is therefore not intended that this invention be limited, except as indicated by the appended claims. It is also intended that the word "network" as used in the specification and claims be read to cover any communication system unless the context requires otherwise and likewise "end system" be read to encompass any suitable end system (telephone, television) on any such communication system or to encompass distributed points in the network intermediate of an end systems. It is also intended that the word "packet" as used in the specification and claims be read to cover any unit of transmitted data, whether an ethernet packet, a cell, or any other data unit transmitted on a network unless the context requires otherwise.

Other Reference Publication (8):
Schwager, "Remote Network Monitoring MIB," Annual Review of Communications, National Engineering Consortium, Chicago, IL, vol. 46, Jan. 1992, pp. 752-754.

CLAIMS:

6. The method according to claim 4 wherein said compiled statistics are as defined by published RMON or RMON2 monitoring protocols.

9. The method according to claim 1 wherein said collector communicates with said network manager using a first protocol, said first protocol being a higher layer protocol defined for the monitoring and management of networks and wherein said node communicates with said collector using a second protocol, said second protocol being a lower layer protocol that in unacknowledged and is specifically designed for lower layer network management communication.

23. The method according to claim 19 wherein said collector communicates with said network manager using a first protocol, said first protocol being a higher layer protocol defined for the monitoring and management of networks and wherein said node communicates with said collector using a second protocol, said second protocol being a lower layer protocol that is flexibly either unacknowledged or acknowledged, has low overhead, and is specifically designed for lower layer network management communication.

Previous Doc Next Doc Go to Doc#

L2: Entry 3 of 3 File: USPT Aug 18, 1998

DOCUMENT-IDENTIFIER: US 5796721 A

**** See image for Certificate of Correction ****

TITLE: Method and system for monitoring fieldbus network with dynamically alterable packet filter

Abstract Text (1):
An improved system and method for monitoring a fieldbus network. The improved method and monitor utilize multiple filters with the capability of simultaneously capturing packets from more than one fieldbus and the ability to apply multiple filters to any single fieldbus. Filtered packets are captured as capture documents and stored in the monitor's memory storage. Filtered packets can be displayed, in real time, on the monitor's display screen. The improved monitor is configured to perform post-capture filtering of captured packets. Post-capture filtering does not destroy data. The improved monitor permits dynamic altering of filter settings. Using this feature, the user can initiate capture using a first filter settings, alter the filter setting while packets are being captured, and apply the altered filter setting to the fieldbus without terminating capture. The altered filtered settings are applied to the fieldbus substantially instantaneously and the packets captured under the altered filter settings are displayed.

Brief Summary Text (2):
The invention relates to the field of network monitors or bus monitors. In particular, the invention relates to a fieldbus network monitor for monitoring information from multiple fieldbuses simultaneously, applying multiple filters to the packets on each fieldbus, dynamically altering filters during capture, displaying packets in real time and in a readable format that assigns a unique color to each layer of the packets, and post-capture filtering of packets.

Brief Summary Text (4):
A fieldbus is a specific type of local area network (LAN that is used to monitor or control one or more pieces of production equipment. A fieldbus network comprises a plurality of digital devices and control/monitoring equipment that are integrated to provide I/O and control for automated processes. A fieldbus network is typically used in industrial and/or process control application, such as a factory or manufacturing plant. The physical devices in a fieldbus system are connected by the fieldbus.

Brief Summary Text (5):
Fieldbus networks may contain one of four types of devices, these being temporary devices, field devices, interface devices, and monitor devices. Temporary devices are devices attached to one of four network addresses reserved for temporary or visitor use. Temporary devices are typically used for configuration and troubleshooting. Field devices are devices that contain function block application processes or, in other words, devices that perform the I/O and control that automates the plant or factory. All field devices are given a permanent address by the system manager when they are attached to the network. Interface devices perform data display and other interface functions for field devices. Like field devices, interface devices are assigned a permanent address, but interface devices do not necessarily contain function block application processes. Finally, monitor devices are devices that are able to listen to network traffic but are not permitted to transmit onto the network. Monitor devices receive no address when attached to the network, and the other network devices are unaware of the monitor's presence.

Brief Summary Text (9):
To detect, analyze, and debug network errors, system designers use network monitors. Fieldbus

monitors generally comprise a computer including bus monitoring circuitry that is electronically connected to the network and which monitors or listens for fieldbus packets on the fieldbus . Fieldbus monitor devices passively interact with the network, listening to network traffic but unable to transmit information onto the fieldbus. Early versions of network monitors did not provide for real time error detection. Instead, packets captured by the computer were saved into an electronic storage medium and subsequently analyzed. Because real time error detection is critical in minimizing the loss of information and the malfunctioning of equipment, monitors lacking real time error detection were of limited value.

Brief Summary Text (10):
Subsequent fieldbus monitors, such as the monitor disclosed in U.S. Pat. No. 5,442,539 to Crowder et al. (hereinafter referred to as "Crowder"), include real time error detection but contain numerous other drawbacks which limit their utility. For example, conventional fieldbus monitors do not permit the user to save captured packets for subsequent review and analysis. Instead, the packet information is merely displayed on the display screen. Once the data leaves the display screen, it is gone. Further, prior art fieldbus monitors do not provide for simultaneously capturing packets from more than one bus or applying multiple filters to each bus. Instead, conventional fieldbus monitors initiate the capture of packets from a single bus and apply a single data filter to the packets. Because fieldbus packets contain multiple layers of information, the single filter restriction in prior art monitors is a significant limitation.

Brief Summary Text (11):
For example, the user of a conventional prior art fieldbus monitor cannot isolate and display FMS layer information from a string of packets while simultaneously isolating and displaying FDL layer information from the same packet string in a different display window. Using a conventional single filter monitor, the user would be forced to filter both the FDL and the FMS layers from the packets, capture both layers into a single capture document, and display both layers together. As another example, a fieldbus user may wish to view and capture FDL information from fieldbus address X while simultaneously viewing and capturing FMS information from fieldbus address Y. As in the first example, conventional fieldbus monitors will not be able to meet the users requirements because such monitors lack the capacity to apply multiple filters to the bus and because conventional monitors lack the capability to filter packets based upon the fieldbus address from which the packet originated.

Brief Summary Text (12):
In addition, if the receiver decides to change or adjust filters, the user is required to discontinue capture, change the filter and then continue. This results in an inability to capture and analyze packets during the time that the filter is being changed. Compounding the problems associated with the single filter, conventional monitors do little to enhance the readability of the information contained in the filtered packets. Fieldbus packets are difficult to read under the best of circumstances because the multiple layers of information contained within each packet are not easily distinguished from one another.

Brief Summary Text (13):
Moreover, the single filter that does exist in conventional fieldbus monitors can only be invoked during the capture of data. In such monitors, the packet filter is initiated prior to capture, capture is initiated, and the filtered information is captured and displayed. The information that is not passed through the filter is irretrievably lost. The user cannot subsequently decide that it would be useful to view information that was not selected in his original filter because there is no mechanism for doing so. A monitor that enabled the user to perform filtering operations on data that was previously captured would improve the monitor's utility significantly. With such a "post filtering" monitor, the user could choose to capture all data, assuming that memory size is not a constraint, and perform all of the desired filtering post capture.

Brief Summary Text (14):
Conventional fieldbus monitors also cannot dynamically select filter settings. Instead, the user must initialize the single filter prior to data capture. During data capture, the user is prohibited from altering filter settings. As a result, if the user desires to change the filter settings, he or she must first terminate the capture of data. Aside from the obvious inconvenience, this unnecessary limitation is significant because a user who has discovered a network error or an otherwise interesting packet sequence might wish to change the filter setting without terminating the process that is executing. If the user is forced to halt the

system, he or she may be unable to recreate the particular condition thereby increasing the time to debug or troubleshoot the network.

Brief Summary Text (16):
The present invention comprises an improved bus monitor system and method for monitoring a fieldbus network. The improved method and monitor utilize multiple filters with the capability of simultaneously capturing packets from more than one fieldbus and the ability to apply multiple filters to any single fieldbus. Filtered packets are captured as capture documents and stored in the monitor's memory storage. Filtered packets can be displayed, in real time, on the monitor's display screen. The filter settings and display settings are manipulated with a user friendly graphical interface.

Brief Summary Text (17):
For each filter window, the user selects the fieldbus packet information layers and/or the fieldbus addresses that are to be monitored with that filter. For example, the user can monitor a fieldbus with a first filter that filters FMS layer information at a first address and a second filter that simultaneously filters FDL layer information at a second address. Both filter windows can be simultaneously displayed on the display terminal. Using the FMS layer filter settings, the FDL layer filter settings, and the network address filter setting, the user can precisely specify the information to be displayed.

Brief Summary Text (18):
The improved monitor is configured to capture filtered packets into storage and to perform post-capture filtering of captured packets. As a simple example, a user may filter and capture all available FDL information from the packets on a fieldbus regardless of the network address. If, after the user has captured packets into a capture document using this filter, he or she desires to isolate and view information originating from a specific network address, the post-capture filtering capability of the monitor allows the user to do so. Post-capture filtering does not destroy data. Thus, after performing a first post-capture filtering, the user could then alter the post-capture filter and perform a second post-capture filter on the captured data.

Brief Summary Text (19):
The improved monitor permits dynamic altering of filter settings. Using this feature, the user can initiate capture using a first filter settings, alter the filter setting while packets are being captured, and apply the altered filter setting to the fieldbus without terminating capture. The altered filtered settings are applied to the fieldbus substantially instantaneously and the packets captured under the altered filter settings are displayed.

Brief Summary Text (20):
Broadly speaking, an improved method of monitoring a fieldbus comprises initializing one or more packet filters. The one or more packet filters are concatenated to form a single filter applied to the data. Fieldbus packets traveling over the fieldbus network are received by a receptor and routed to the filter. The filter filters the received packets the data indicated by each of the one or more packet filters. The filtered data is then separated based on each of the filters and is then stored as captured packets. The filtered data for each packet filter is then sent to a display generator for displaying on a display screen.

Brief Summary Text (21):
The improved method of monitoring a fieldbus further comprises, applying a post-capture filter to the captured packets to produce twice filtered packets which can be displayed.

Brief Summary Text (22):
An improved apparatus for monitoring packets comprises a packet receptor which is connected to receive packet filter information, wherein the packet filter information comprises a combination of one or more user-selected packet filters. The packet filter produces filtered data as an output, which is then separated according to the one or more user-selected filters. The filtered data for each user-selected packet is then routed to a storage means for capture. In addition, a display generator is connected to the output of the filter packets for real time displaying of the captured information.

Drawing Description Text (5):
FIG. 3 is a prospective view of one embodiment of an improved fieldbus monitor;

Drawing Description Text (6):
FIGS. 4A and 4B are a block diagrams of one embodiment of an improved fieldbus monitor;

Drawing Description Text (16):
FIG. 14 is the flow diagram of an improved method for monitoring a field bus;

Detailed Description Text (3):
Turning now to the drawings, FIG. 1 is a block diagram of fieldbus network 10. System 10 is comprised of a fieldbus 16 and a plurality of fieldbus devices 12 connected to fieldbus 16. An improved monitor 14 according to the present invention is connected to the fieldbus 16. Fieldbus devices 12, monitor 14, and fieldbus 16 are compliant with the Fieldbus Specification published by the Fieldbus Foundation. Fieldbus devices 12 are capable of generating fieldbus packets on fieldbus 16. Each fieldbus device 12 is assigned a permanent network address. Fieldbus monitor 14 passively interacts with fieldbus 16 to monitor packets on the fieldbus. In the preferred embodiment, fieldbus monitor 16 does not provide packets on fieldbus 16.

Detailed Description Text (5):
Each fieldbus packet may contain data from each layer in protocol stack 17. Separating the various fieldbus protocol layers from one another in the packet can enhance the readability of displayed information and eliminate unwanted layers from the display. As discussed in greater detail below, the improved fieldbus monitor of the present invention is capable of separating individual fieldbus protocol layers and displaying information from the various layers in readable format.

Detailed Description Text (6):
FIG. 3 is a perspective view of an improved fieldbus monitor according to the present invention. Monitor 14 is comprised of a computer 40 having a display screen 41, a keyboard 48 and an alternative input device 50, such as a "mouse" or other pointing device. The computer 40 includes a field bus device or card 13 which performs fieldbus interface functions. Fieldbus card 13 is comprised in computer system 40, but is shown external to the computer system 40 for illustrative purposes. The computer 40 of monitor 14 comprises various standard components, including computer storage or memory 46 and a computer processor 44 (shown in FIG. 4). In a preferred embodiment of monitor 14, the computer is a personal computer with a keyboard and a mouse running under the Windows.RTM., Windows 95.RTM., or Windows NT.RTM. operating systems.

Detailed Description Text (7):
Turning now to FIG. 4, a block diagram of fieldbus monitor 14 is shown. Fieldbus monitor 14 comprises fieldbus card 13 which interacts with computer 40 via ISA bus 41. Fieldbus card 13 has N ports, each connected to a separate fieldbus, for receiving packets from N separate buses. Each port includes a packet receptor (30A, 30B, . . .,3 ON) and a mechanism for time stamping and decoding each arriving packet (32A, 32B, . . .,32N). After the packets have been time stamped and decoded, they are filtered with the concatenated filters 34A, 34B, . . .,34N. As discussed further below, one or more filter windows per port (i.e. per bus) may be selected by the user. The user defines the information that appears in each filter window by assigning a combination of protocol layers and network addresses to each filter window. All of the filter window settings for a given port are then logically "OR'ed" together to create a single concatenated filter (per port) which is downloaded to fieldbus card 13. The concatenated filter 34N, shown in the exploded view of FIG. 1, demonstrates the concatenation feature. The concatenated filter is formed by "OR'ing" the various protocol layers that the user has specified and "AND'ing" the network addresses that the user has specified. For example, consider a user who wants to monitor packets on the fieldbus connected to port 1. Suppose that the user wants to view the FDL protocol layer information from network address 100 in one filter window and the FMS protocol layer information from address 50 in a second filter window. The user would select the appropriate settings from the filter settings window. The monitor would then concatenate the filters to produce a single concatenated filter for port 1. As packets are subsequently received at port 1, the concatenated filter produces filtered packets by eliminating all packet information not associated with the specified protocol layers and not originating from the specified network addresses. The filtered packets are routed to a buffer within fieldbus storage unit 39 and subsequently delivered to computer 40 over ISA bus 41 for analysis and display.

Detailed Description Text (8):
FIG. 5 is a block representation of the computer display of fieldbus monitor 14. Using the multiple window display capabilities of commercially available operating systems such as

Microsoft Windows.RTM., monitor 14 creates separate filter windows 54 on computer display screen 41 for each packet filter 56. In the example shown in FIG. 5, four different filter windows (54a, 54b, 54c, and 54d) are being displayed on display screen 52 of computer 40. Each filter window 54 is associated with a corresponding packet filter 56. As shown in the figure, the packet information displayed on the screen is generated from the capture documents 43A and 43B. Capture documents 43 contain information filtered from incoming packets using the concatenated filters referred to above. Monitor 14 processes and separates the filtered packets in capture documents 43 into different filter windows 54 according to the packet filters 56. The information associated with each packet filter 56 is displayed on display screen 52 in a corresponding display window 54. The format in which monitor 14 displays the information in each filter window 54 can vary according to user selections described in more detail below with respect to FIG. 10. FIG. 5 also demonstrates the ability of monitor 14 to display packets captured from more than one fieldbus. Each capture document 43 is associated with a different fieldbus 16.

Detailed Description Text (9):
Turning now to FIGS. 6, 7, and 8, the available settings for each packet filter 56 are shown. As shown in the figures, filter display 70 is comprised of three separately selectable filter layers. FIG. 6 shows FDL filter layer setting 72. An improved monitor 14 hereof can select which FDL packet types to display or capture. By simply filling in or checking the boxes in the packet types desired, the user can easily select those FDL packet types he or she wishes to view. In the example shown in FIG. 6, all FDL packet types have been selected. Similarly, FIG. 7 shows the FMS filter layer settings interface 74 and FIG. 8 shows the address filter interface 76. Because of the large number of FMS services that monitor 14 can filter, FMS layer interface 74 divides the various services into six FMS PDU types as shown in FIG. 7. The user selects the desired PDU types from PDU selection box 73 and checks the desired services in services selection box 75. As its name implies, address filter 76 filters out packets originating only from specified fieldbus network addresses. Address filter interface is useful in screening large numbers of packets when the user's interest is in certain addresses only. During the capture of information, the three different filter layers are logically joined. If no addresses are selected in address filter interface 76, monitor 14 "OR's" the FDL layer filter and the FMS filter. If an one or more addresses are selected in address filter 76, monitor 14 screens incoming packets for the selected address prior to passing the packets through the FDL and FMS filters.

Detailed Description Text (10):
FIG. 9 depicts the fieldbus settings interface 80 used to control capture, display, and toolbar settings. Using capture-setting interface 82, the user enters the board name from which he or she desires to capture information in board selection window 83. The improved monitor 14 hereof is capable of capturing and displaying fieldbus packets from multiple fieldbuses simultaneously. A separate fieldbus board 13 (or a board supporting multiple fieldbus channels) must be installed into monitor 14 for each separate fieldbus.

Detailed Description Text (11):
FIG. 10 shows a multi-paned display of a single filter window. Monitor 14 is capable of displaying packet information in a variety of formats regardless of the filter used. In a preferred embodiment of the monitor 14, each filter window 54 may be split into as many as four filter window panes, shown in the drawing as 90a, b, c, and d. Filter window panes 90a, b, and c are packet views of the filtered packets. Packet view displays filtered information as a series of packets on the display screen. Using packet view selector 92, the user can choose one of three packet view formats for display. Filter window pane 90b shows the hex packet view format in which packets are displayed in hexadecimal format. Filter window pane 90a, shows filter packets being displayed using the decode packet view in which the packet information is decoded into its instructions. Filter packet window pane 90c, represents the simple packet view in which only the FDL PDU type is displayed. Filter window pane 90d, shows the filtered information being displayed in statistics view.

Detailed Description Text (12):
As shown in greater detail in FIG. 11, the statistics view shows the percentage of time or the amount of bandwidth used by each layer of the fieldbus protocol stack. In a preferred embodiment of monitor 14, the statistics view, displays information in as many as three different pie charts. Pie chart 100a, the band width pie chart, shows the percentage of the fieldbus bandwidth that the various fieldbus layers are consuming. Pie charts 100b and 100c further break down the FMS PDUs and the FDL PDU's to FMS Services and FDL PDU types

respectively. Pie charts 100b and 100c are useful in determining which FMS Services and which FDL PDUs are being used most frequently on the fieldbus. This information can be useful in debugging and analyzing a fieldbus. Returning to FIG. 10, statistics view selector 94 is capable of selecting any of the three statistics view displays.

Detailed Description Text (13):
FIG. 12 shows the display settings interface 110 of a preferred embodiment of monitor 14. Within display settings interface 110, the user can assign unique display colors to each packet layer with color selector box 112. In addition, interface 110 can assign separate colors to unknown information and to the time-of-day information. Especially when viewing packets in packet view, which is shown in greater detail in FIG. 13, assigning separate colors to the various information layers greatly enhances the readability of the display. FIG. 13 shows packets being display in the packet view using the simple mode. In addition to simple mode, in which only the PDU type of the packets are shown, monitor 14 can display packets in hex mode, shown as 90b in FIG. 10, and decode mode, shown as 90a in FIG. 10. Hex mode displays the hexadecimal code of the packet data in addition to the PDU types, while the decode mode translates the hexadecimal data into readable English. Although not easily seen in the black and white hardcopy of FIG. 13, the FDL, FMS, and FAS PDU are uniquely colored in accordance with monitor 14 hereof.

Detailed Description Text (14):
Turning now to FIGS. 14 through 16, flow diagrams for an improved method of monitoring a field bus are shown. In FIG. 14, a method is shown for monitoring a field bus comprising initializing the settings for a first packet filter in step 102. Initializing a packet filter is accomplished by selecting any permutation of the information layer packets and network addresses described above in reference to FIGS. 6-10. If the user desires to use the multiple filter capability of the present invention, filter settings, preferably unique settings, are selected for each desired filter. The blank space between step 102 and step 104 in FIG. 14 represents this iterative process. In step 104, the "Nth" or final filter is set.

Detailed Description Text (15):
After all of the filter settings have been selected, the settings from the multiple filters are concatenated or combined (or ORed) together to create a single filter setting comprising a union of each individual filter setting. As a simple example, suppose the user desires a first filter to filter all FDL layer packets originating from fieldbus network address 32 and a second filter to filter all FMS packets originating from fieldbus network address 196. The user would open up a first packet filter dialog box, click on the "Check All" button of the FDL Filter menu shown in FIG. 6, and select address 32 from FIG. 8. The user would then open up a second packet filter dialog box, click the "Check All" button of the FMS Filter menu shown in FIG. 7, and select address 196. The improved fieldbus monitor then concatenates or combines these two packet filters in step 106 and downloads the single concatenated filter to the fieldbus card 13 in step 108. The single concatenated filter is then applied to the packets on the fieldbus network 16 to filter all FDL layer packets originating from address 32 and all FMS layer packets originating from address 196. This filtering step is shown as step 110 in FIG. 14. The improved fieldbus monitor then separates the packets associated with the first packet filter settings from the packets associated with the second packet filter settings in step 1 12. Thereafter, the improved monitor displays the first filter packets in a first display window and the second filter packets in a second display window as shown in step 114. The filtered packets could also be captured, as shown in step 116, by storing the filtered packets in fieldbus storage.

Detailed Description Text (16):
FIG. 15 is a flow diagram of one embodiment of the present invention illustrating the improved fieldbus monitor's ability to dynamically alter the filter settings. The dynamic altering of filters can occur at any time that the fieldbus monitor is filtering, capturing, and/or displaying packets as described above in reference to FIG. 14. To initiate the dynamic alteration process, the user simply edits the settings on the menus filter dialog box described above in reference to FIGS. 6-10. Once the user has completed altering the settings for a particular filter in step 118, the improved fieldbus monitor receives the updated packet information in step 120 and makes the necessary alterations to the single concatenated filter as shown in step 122. The improved fieldbus monitor then downloads the altered filter settings to the fieldbus card in step 108 and fieldbus monitoring continues as described above in reference to FIG. 14.

Detailed Description Text (17):
The present invention handles the dynamic altering of the filter settings "on the fly" such that it is unnecessary to interrupt the capturing of packets. With each new packet filter setting selection, a new filter is downloaded to fieldbus board 13. The improved fieldbus monitor handles the altered filter settings similar to the manner in which interrupts are handled by a microprocessor.

CLAIMS:

1. A method of monitoring fieldbus packets on a fieldbus network using a monitor configured on the fieldbus network to receive the fieldbus packets, wherein the monitor includes a display screen, the method comprising:

initializing a first packet filter;

filtering said fieldbus packets with said first filter to produce filtered data;

capturing said filtered data, wherein said capturing includes storing said filtered data in a memory of said fieldbus monitor; and

dynamically altering said first filter to produce an altered filter, wherein said fieldbus packets filtered after said dynamically altering are filtered with said altered filter and, wherein said dynamically altering occurs without substantially interrupting said filtering.

2. The method of claim 1 further comprising:

displaying in real-time said filtered data on said display screen of said monitor.

4. The method of claim 3, further comprising displaying in real-time said filtered data on said display screen of said monitor wherein a unique display color is associated with each of said plurality of information layers.

5. The method of claim 1 further comprising:

terminating said capturing of said filtered data;

modifying said altered filter to produce a post-capture filter; and

further filtering said filtered data stored in said memory of said fieldbus monitor with said post-capture filter.

7. An apparatus for monitoring fieldbus packets on a fieldbus network, said fieldbus packets each comprising a plurality of information layers and originating from one of a plurality of network addresses, said apparatus comprising:

a packet receptor having an input and an output, said input coupled to said fieldbus network;

means for initializing a first packet filter, wherein said first packet filter corresponds to a permutation of said plurality of information layers and network addresses;

filter logic including an input connected to an output of said packet receptor, wherein said filter logic uses said first packet filter to filter the fieldbus packets on the fieldbus network;

means for dynamically altering said first packet filter to produce an altered filter for filtering packets wherein said altering occurs without substantially interrupting said packets; and

a memory coupled to said filter logic for storing said filtered packet data.

Previous Doc Next Doc Go to Doc#